REMARKS

Claims 17-20 are new; thus, claims 1-20 are all the claims pending in the application. Claims 1-16 stand rejected on prior art grounds. Applicants respectfully traverse these rejections based on the following discussion.

I. The Prior Art Rejections

Claims 1-16 stand rejected under 35 U.S.C. §102(e) as being anticipated by Underwood (U.S. Patent No. 6,609,128). Applicants respectfully traverse these rejections based on the following discussion.

The claimed invention provides a method for determining the possibility of an adverse effect arising from a code change in a computer program. For a given code change to an important class, the method runs test cases associated with the important class and dependent classes of the important class. In the rejection, the Office Action argues that the prior art of record discloses many features of the claimed invention. However, Applicants submit that the "testing" in Underwood is not performed when there is a code change in a "business component" (which the Office Action asserts teaches the "important classes" of the claimed invention). Instead, the "testing" in Underwood is merely performed after the "business components" are defined. In other words, the "testing" in Underwood is performed whether or not there is a code change to the "business components"; Underwood does not mention code changes to the "business components" or running test cases for such code changes. Therefore, as explained in

greater detail below, Applicants respectfully submit that the prior art of record does not teach or suggest the claimed invention.

Applicants traverse the rejections because Underwood fails to disclose the claimed features of associating test cases with said important classes and with said directly and indirectly dependent classes; and for a given code change to a first important class: running all test cases associated with the first important class and associated with dependent classes of the first important class. Such features are defined in independent claims 1, 6, 11, 15, and 16 using similar language.

The Office Action argues that the "business components" and "functional interrelationships" of Underwood teach the "important classes" and the "directly and indirectly dependent classes" of the claimed invention. Specifically, on page 3 of the Office Action, items 14502 and 14504 (FIG. 145.1) and the accompanying text of Underwood are cited to reject the claimed "business components" and "functional interrelationships".

First of all, Applicants submit that the "functional interrelationships" of Underwood are not separate and distinct components that are dependent on the "business components". Instead, the "functional interrelationships" of Underwood merely describe the way in which the "business components" (which the Office Action asserts teaches the "important classes" of the claimed invention) are related (See Underwood, FIG. 145.1, items 14502 and 14504). Therefore, it is Applicants' position that contrary to the position taken in the Office Action, the "functional interrelationships" of Underwood do not teach the "dependent classes" of the claimed invention.

Further, the Office Action asserts that it would be inherent to: first, associate test cases with the "business components" and "functional interrelationships" of Underwood; and, second, run the associated test cases for a given code change to a "business component". Specifically, the Office Action asserts that such associating and running of the test cases would be inherent because Underwood discloses "testing the functional aspects of the code modules" (Office Action, p. 3 (citing Underwood, FIG. 145.1, items 14506 and 14508)). Applicants respectfully disagree.

More specifically, Applicants submit that the "testing" in Underwood is not performed when there is a code change in a "business component" (which the Office Action asserts teaches the "important classes" of the claimed invention). Instead, the "testing" in Underwood is merely performed after the "business components" are defined and the "functional interrelationships" are identified (Underwood, FIG. 145.1). In other words, the "testing" in Underwood is performed whether or not there is a code change to the "business components". Applicants submit that Underwood does not discuss code changes to the "business components" or running test cases for such code changes.

Additionally, Applicants submit that the "testing" performed in Underwood does not run test cases for the "business components" and "functional interrelationships" (which the Office Action asserts teaches the "important classes" and the "directly and indirectly dependent classes" of the claimed invention). Instead, the "testing" performed in Underwood tests the functionality of "code modules", which are generated to carry out the capabilities of the "business components" and "functional interrelationships". Nevertheless, testing the "functionality" of the "code modules" of Underwood does not

run test cases associated with the "business components" and "functional interrelationships".

More specifically, as described in column 296, lines 6-21 of Underwood, FIG. 145 illustrates a method 14500 for generating software based on business components. A plurality of business components in a business are defined in operation 14502 with each business component having a plurality of capabilities. In operation 14504, functional interrelationships are identified between the business components. Code modules are generated in operation 14506 to carry out the capabilities of the business components and the functional interrelationships between the business components, while ensuring the capabilities that are carried out by each code module are essentially unique to the business component associated with the code module. In operations 14508 and 14510, the functional aspects of the code modules and the functional relationships of the code modules are tested. The code modules are subsequently deployed in an e-commerce environment in operation 14512.

To the contrary, as described in paragraph 0024-0025 of Applicants' disclosure, the test case or cases for each class are now defined (step 24). This involves specifying a set of steps to be performed and the expected results at each step. The authors of such test cases are skilled programmers, and the nature of the test cases depends upon the software high level specifications. In execution, if all the steps give the expected results, then the test case is considered to be successful. The test cases are associated with the "important" and dependent classes (step 26). Now, with reference to FIG. 3, when the code for a particular class is changed (step 30), the test case or cases associated with it are run (step

32). The dependent (i.e. both direct and indirect) classes for this type are also found (step 34), and the associated test cases are run (step 36). If a class is not important, then there will not be any associated test case to be run.

Accordingly, Applicants submit that the "testing" in Underwood is not performed when there is a code change in a "business component" (which the Office Action asserts teaches the "important classes" of the claimed invention). Instead, the "testing" in Underwood is merely performed after the "business components" are defined (Underwood, FIG. 145.1). In other words, the "testing" in Underwood is performed whether or not there is a code change to the "business components"; Underwood does not mention code changes to the "business components" or running test cases for such code changes.

Furthermore, Applicants submit that the "testing" performed in Underwood does not run test cases for the "business components" and "functional interrelationships" (which the Office Action asserts teaches the "directly and indirectly dependent classes" of the claimed invention). Instead, the "testing" performed in Underwood tests the functionality of "code modules", which are generated to carry out the capabilities of the "business components" and "functional interrelationships". Additionally, Applicants submit that the "functional interrelationships" of Underwood are not separate and distinct components that are dependent of the "business components". Instead, the "functional interrelationships" of Underwood merely describe the way in which the "business components" are related.

Therefore, it is Applicants' position that Underwood fails to disclose the claimed features of determining directly and indirectly dependent classes of said important classes; associating test cases with said important classes and with said directly and indirectly dependent classes; and for a given code change to a first important class: running all test cases associated with said first important class and associated with dependent classes of said first important class as defined in independent claims 1, 6, 11, 15, and 16. Further, it is Applicants' position that dependent claims 2-5, 7-10, 12-14, and 17-20 are similarly patentable, not only because of their dependency from a patentable independent claims, but also because of the additional features of the invention they defined. In view of the foregoing, the Examiner is respectfully requested to reconsider and withdraw the rejections.

II. Formal Matters and Conclusion

In view of the foregoing, Applicants submit that claims 1-20, all the claims presently pending in the application, are patentably distinct from the prior art of record and are in condition for allowance. The Examiner is respectfully requested to pass the above application to issue at the earliest possible time.

Should the Examiner find the application to be other than in condition for allowance, the Examiner is requested to contact the undersigned at the local telephone number listed below to discuss any other changes deemed necessary. Please charge any deficiencies and credit any overpayments to Attorney's Deposit Account Number 09-0441.

Respectfully submitted,

Dated: January 9, 2008

/Frederick W. Gibb, III/
Frederick W. Gibb, III
Registration No. 37,629

Gibb & Rahman, LLC
2568-A Riva Road, Suite 304
Annapolis, MD 21401
Voice: (410) 573-1545
Fax: (301) 261-8825
Customer Number: 29154